Cutting plane reusing methods for multiple dual optimizations

Kouhei Harada *

February 9, 2023

Abstract

We consider solving a group of dual optimization problems that share a core structure: Every primal problem of the group is obtained by the right-hand side variation of constraints in the original primal problem, while the other core part of the original primal problem, such as the objective and the left-hand side of the constraints, are kept fixed.

Since evaluations of the dual function are usually expensive in dual optimization, many algorithms are designed to reduce evaluations. However, they are tailored to solve one dual problem. We propose a framework of algorithms based on cutting plane methods to reduce total evaluations for multiple dual optimization problems.

In the proposed algorithms, we construct a tree of evaluation points in which a route from the root node to a leaf node corresponds to a sequence of evaluation points for one dual problem, and the root node corresponds to an initial point that is common for all dual problems. Since essential parts of the evaluations are invariable among the group of dual problems, we can reuse the information. Our numerical experiments endorse the efficiency of the proposed algorithms.

Key words. dual optimization, cutting plane, Lagrangian relaxation, analytic center

AMS subject classifications. 90C22, 90C25

1 Introduction

We are concerned in this paper with a group of optimization problems. Let T be the number of the group of optimization problems, and $t = 1, \cdots, T$ be an index corresponding to each optimization problem. Their original primal optimization problems are given as follows:

$$\begin{array}{rl} \text{minimize} & f(x) \\ \text{subject to} & g(x) \le u, \\ & x \in X, \end{array} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, $g_j : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, $j = 1, \cdots, m$ are common functions, X is a common closed set, and $b \in \mathbb{R}^m$ is a variable parameter. We assume that the structure of (f, g, X) is unchanged among the group of optimization problems. On the other hand, we assume that the parameter u is variable for each optimization problem indexed by t. The sequence of the parameters for each t is given by $\{u_t\}_{t=1}^T$. By relaxing the constraints (1.1) with a positive dual vector $y \ge 0$, one can obtain the dual problems as

$$\begin{array}{rl} \text{maximize} & \min_{x \in X} \{f(x) + \langle y, g(x) \rangle\} - \langle u, y \rangle \\ \text{subject to} & y \ge 0. \end{array}$$

We assume that each of the dual problems corresponding to index t has an optimal dual solution y_t^*. Then, there exists a sufficiently large positive m dimensional vector $\bar{y} > 0$ such that $0 \le y_t^* \le \bar{y}$ for all

*NTTDATA Mathematical Systems Inc, 1F, Shinanomachi Renga-Kan, 35, Shinanomachi, Shinjuku-ku, Tokyo, JAPAN, email:harada@msi.co.jp

$t = 1, \cdots, T$. To simplify the argument, we transform the dual problems into bounded and convex as follows.

$$\begin{array}{ll} \text{minimize} & q_u(y) \\ \text{subject to} & 0 \le y \le \bar{y}, \end{array}$$

where $q_u(y)$ be a combination of a convex function $q(y)$ and linear function $\langle u, y \rangle$ defined as

$$q_u(y) \equiv q(y) + \langle u, y \rangle, \quad q(y) \equiv -\min_{x \in X} \{f(x) + \langle y, g(x) \rangle\}.$$

We denote the form of dual problems as $\mathcal{D}(u)$. We stress that the structure of $q(y)$ is stable among the group of dual problems because we assume that (f, g, X) is unchanged.

1.1 Basic ideas

Our goal in this paper is to solve a group of dual problems $\{\mathcal{D}(u_t)\}_{t=1}^T$ efficiently. Since dual optimization problems are non-smooth convex problems, one usually uses an algorithm for non-smooth convex optimization, such as subgradient methods, cutting plane methods, and bundle methods [21, 13, 3, 25]. However, these known algorithms are designed to solve one optimization problem. In this paper, we need to solve two or more dual optimization problems with a common structure.

Since the central part of the objective $q(y)$ is unchanged, we can permanently use the information throughout solving the group of dual optimization problems. From the viewpoint of reusing the information of $q(y)$, cutting plane methods are more attractive than subgradient methods. In subgradient methods, one can reuse only one subgradient obtained near the optimal solution as a restarting point. On the other hand, in cutting plane methods, one can reuse many cutting planes obtained so far to reduce the region that includes the optimal solution. Bundle methods can also reuse the information. Readers may know that bundle methods can be seen as an elaborated variation of cutting plane methods [21]. However, we focus on cutting plane methods in this paper to clarify our idea.

In general, not all the cutting planes obtained so far are useful for different objectives. Moreover, if we consider all cutting planes, the size of the subproblem quickly diverges. With this in mind, we propose a framework of methods that reuse efficiently chosen cutting planes. We call the proposed methods *cutting plane reusing methods*. As a cutting plane method generates a sequence of evaluation points, a cutting plane reusing method generates a tree of evaluation points. In the tree, each node corresponds to an oracle point y and a subset of indices t. The root node corresponds to a common initial point and the set of all indices $\{t\}_{t=1}^T$. A leaf node corresponds to an optimal solution for some index t and its corresponding singleton $\{t\}$. Therefore, a route from the root node to a leaf node corresponds to a sequence of evaluation points for a dual optimization problem.

To intuitively understand the tree structure, we give a simple example that consists of three dual optimization problems. Figure 1 illustrates the tree structure where y_S^k denotes the k-th common evaluation point for all indices $t \in S$. In this example, first and second evaluation points $y^1_{\{1,2,3\}}$ and $y^2_{\{1,2,3\}}$ are commonly used for all indices $t = 1, 2, 3$, and third evaluation point $y^3_{\{2,3\}}$ is commonly used for $t = 2, 3$.

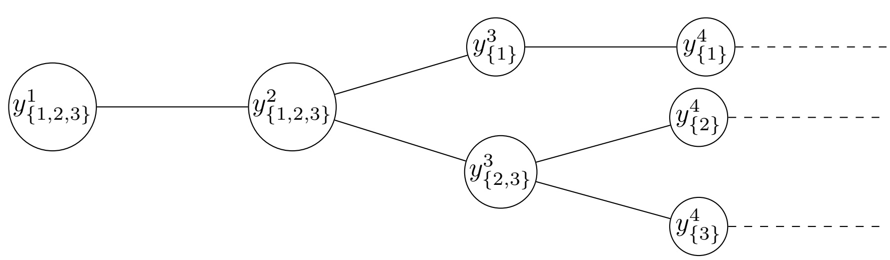

Figure 1: Tree of evaluation points.

1.2 Relation to warm-start

The idea of reusing the past information itself is not novel in optimization. For example, warm-start is a well-known technique that also reuses past information. However, there are several essential differences. First, warm-start only use the information on an optimal solution of the previous problem. On the other hand, the proposed algorithm makes use of the information on many previous evaluation points. Second, warm-start is effective only when the optimal solution of the previous problem is close to an optimal solution in the current optimization problem. However, the proposed algorithm is effective even if the optimal solutions among the optimization problems are not close. Furthermore, the proposed algorithm is designed to solve multiple optimization problems simultaneously, which is different from the original scope of warm-start.

1.3 Relation to other optimization problems

Our goal is to solve a group of dual optimization problems efficiently. One may think that the goal is somewhat similar to a multiobjective optimization problem in which the objective is defined as a T dimensional vector function $(q_{u_1}(y), \cdots, q_{u_T}(y))$. The aim of multiobjective optimization problems is obtaining the Pareto frontier, or a Pareto optimal solution [6]. The weighted sum method [30] is a popular method to solve multiobjective optimization problems. In our setting, the weighted sum method recursively minimizes $\sum_{t=1}^{T} w_t q_{u_t}(y)$ for a parametric weighted vector w such that $\sum_{t=1}^{T} w_t = 1$ and $w_t \geq 0$. Therefore, solving subproblems in the weighted sum method for $w = e_1, \cdots, e_T$, where e_t is a unit vector t-th element of which is 1, is equal to our goal. However, our goal is different from that of the multiobjective optimization problem. We need neither the Pareto frontier nor a Pareto optimal solution. Our problem is much simpler than the multiobjective optimization problem: We need an optimal solution for each t.

One may think our goal is relevant to a parametric dual optimization problem where u is a parameter. Let us denote the optimal solution set of $\mathcal{D}(u)$ by $Y^*(u)$, i.e, $Y^*(u) \equiv \arg\min_{0 \leq y \leq \bar{y}} q_u(y)$. Then, the parametric optimization problem aims to know how the optimal solution set $Y^*(u)$ changes as the parameter u slightly moves [28]. However, we are not interested in the mathematical or geometric structures of $Y^*(u)$, such as inner or outer continuity for a particular u. Our goal is to reach an optimal solution $y_t^* \in Y^*(u_t)$ for each parameter u_t.

1.4 Outline of the paper

This paper is organized as follows. In section 2, we explain the basic structure of cutting plane methods. We also briefly explain three popular variations of cutting plane methods. However, comparing the superiority among the cutting plane methods is not the scope of the paper. In section 3, we demonstrate the idea and procedure of cutting plane reusing methods. A cutting plane reusing method needs to be combined with a cutting plane method. We also prepare two concrete examples to understand the algorithms in detail. In section 4, we demonstrate several numerical experiments that endorse the efficiency of cutting plane reusing methods. Finally, we conclude the paper in section 5.

2 Cutting Plane Methods

This section briefly reviews the algorithms known as cutting plane methods. We use the term cutting plane methods as a general term for algorithms that use cutting planes to reduce the region iteratively that contains an optimal solution. Confusingly, Kelley's cutting plane method(KCPM) [18] is often also called the cutting plane method. To distinguish them, we call the latter Kelley's cutting plane method. Although cutting plane methods are applicable for general constrained convex optimization problems, we use a dual problem $\mathcal{D}(u)$ as a representative to explain the structure. For more general discussion, see [21, 25]. The polyhedra that consist of the feasible region and the cutting planes are often called *localization sets*. Let k be an iteration count of the cutting plane methods. We denote Ω_k by the localization set of k-th iteration. The outline of cutting plane methods can be described as follows.

Cutting Plane Method

Step 1. (Initialization) Set $k = 0$ and $\Omega^0 = \{ y \in \mathbb{R}^m \mid 0 \le y \le \bar{y} \}$.

Step 2. (Choosing the Next Evaluation point) Choose $y^{k+1} \in \Omega^k$.

Step 3. (Calling an Oracle) Obtain $x^{k+1} \in \operatorname{argmin} \{ f(x) + \langle y^{k+1}, g(x) \rangle \mid x \in X \}$.

Step 4. (Updating the Localization Set) Update $\Omega^{k+1} = \Omega^k \cap \{ y \in \mathbb{R}^m \mid \langle u - g(x^{k+1}), y - y^{k+1} \rangle \le 0 \}$.

Step 5. (Stopping Criteria) If Ω^k is sufficiently small, exit. Otherwise, go to next step.

Step 6. (Updating the Iteration) $k = k + 1$ and goto Step 2.

In view of $u - g(x^{k+1}) \in \partial q_u(y^{k+1})$, the newly generated cut in Step 4 includes an optimal solution. In addition, a compiled form of the localization set is given as follows:

$$\Omega^k = \{ y \in \mathbb{R}^m \mid 0 \le y \le \bar{y},\ \langle u - g(x^\ell), y - y^\ell \rangle \le 0,\ \ell = 1, \cdots, k \} \tag{2.1}$$

Although there are many variations of the cutting plane methods, their main differences emerge in Steps 2 and 4: How to choose the next evaluation point and how to define a localization set.

First, we focus on the choice of the next evaluation point. KCPM constructs a piecewise linear approximation of the objective and chooses a minimizer of the approximation as the next evaluation point. Intuitively, choosing a type of center of the localization set is also attractive. If we choose the center of gravity as the next evaluation point, we recover the method of centers of gravity(MCG)[22]. Another popular choice of a center is the analytic center. If we choose the analytic center as the next evaluation point, we recover the analytic center cutting plane method(ACCPM)[11]. To know other cutting plane methods based on a type of center, such as Maximum volume ellipsoid cutting plane method, Chebyshev center cutting plane method, and Volumetric center cutting plane method, see [12, Section 3][4, Section 4][20, Chapter 4] and the references therein.

Second, we focus on the variation of localization sets. In Step 4, we can easily confirm that the newly generated cutting plane passes the evaluation point y^{k+1}. Such cutting planes are called *central cuts*. On the other hand, it is well known that much tighter cuts below are also possible.

$$\langle u - g(x^{k+1}), y - y^{k+1} \rangle \le \min_{k'=1,\cdots,k+1} q_u(y^{k'}) - q_u(y^{k+1}). \tag{2.2}$$

These cuts are known as *deep cuts*. Clearly, localization sets can be defined by not only central cuts but also deep cuts. Therefore, the compiled form of localization sets with deep cuts is given as follows.

$$\Omega^k_{deep} = \left\{ y \in \mathbb{R}^m \;\middle|\; 0 \le y \le \bar{y},\ \langle u - g(x^\ell), y - y^\ell \rangle \le \min_{\ell'=1,\cdots,\ell} q_u(y^{\ell'}) - q_u(y^\ell),\ \ell = 1, \cdots, k \right\}. \tag{2.3}$$

Defining the localization sets for an epigraph form of the dual problem $\mathcal{D}(u)$ is also possible. Let us define the epigraph form of $\mathcal{D}(u)$ as follows:

$$\begin{array}{ll} \text{minimize} & r \\ \text{subject to} & r \ge q_u(y), \\ & 0 \le y \le \bar{y}. \end{array}$$

In this case, we add two cutting planes in Step 4:

$$q_u(y^{k+1}) + \langle u - g(x^{k+1}), y - y^{k+1} \rangle \le r,\ r \le q_u(y^{k+1}). \tag{2.4}$$

Therefore, the localization sets for the epigraph form is given below.

$$\Omega^k_{epi} = \left\{ (y, r) \in \mathbb{R}^m \times \mathbb{R} \;\middle|\; 0 \le y \le \bar{y},\ r \le \min_{\ell=1,\cdots,k} q_u(y^\ell),\ q_u(y^\ell) + \langle u - g(x^\ell), y - y^\ell \rangle \le r,\ \ell = 1, \cdots, k \right\}. \tag{2.5}$$

Some call the cutting plane methods with epigraph form of localization sets *Epigraph cutting plane methods* [4] to stress the difference. It is straightforward to confirm the following inclusion relation.

$$p(\Omega^k_{epi}) \subseteq \Omega^k_{deep} \subseteq \Omega^k, \tag{2.6}$$

where p is a projection operator that maps (y, r) to y. In the later section of this paper, we use the localization set Ω^k. However, adapting Ω^k_{deep} or Ω^k_{epi} is also possible by slightly modifying the algorithms.

A cutting plane reusing method must use a cutting plane method as a subroutine. As candidates for the subroutine, we choose three popular cutting plane methods: KCPM, ACCPM, and MCG. We use MCG to explain a complicated behaviour of a cutting plane reusing method in Example 3.1 because MCG is easy to handle in one-dimensional cases. We also use KCPM and ACCPM in the numerical experiments to demonstrate the superiority of cutting plane reusing methods, because these two cutting plane methods have been applied for practical problems[23, 10]. There are lots of other cutting plane methods. However, comparing the variations of cutting plane methods is not the scope of the paper. To know other cutting plane methods and their recent advances in detail, see [20, Chapter 4].

2.1 Kelley's Cutting Plane Method

Kelley's cutting plane method approximates the objective as a maximum of linear functions and minimizes the approximate functions to obtain the next evaluation point. Therefore, the optimization subproblem to detect the next point in Step 2 can be described as follows:

$$\begin{array}{ll} \text{minimize} & r \\ \text{subject to} & q_u(y^\ell) + \langle u - g(x^\ell), y - y^\ell \rangle \leq r, \, \forall \ell \in \{1, \cdots, k\}, \\ & 0 \leq y \leq \bar{y}. \end{array} \tag{2.7}$$

We call it the subproblem of KCPM. We can easily confirm that its optimal solution must be a member of the epigraph localization set Ω^k_{epi}. Therefore, relation (2.6) suggests that KCPM is uniformly applicable in Step 2 regardless of the variations of localization sets.

When $k = 0$, the subproblem is unbounded and meaningless. Therefore, we usually choose $y^1 = \bar{y}/2$ as the first evaluation point of the initial localization set Ω^0. Convergence proof of KCPM is straightforward. To confirm it, see [13, Chapter XII, Theorem 4.2.3][3, Theorem 9.6].

KCPM is often used in column generation, because its optimization subproblems are equivalent to the dual of restricted master problems in the context of Dantzig-Wolfe decomposition [7, Section 1]. Indeed, the left-hand side of the constraint (2.7) is equivalent to $f(x^\ell) + \langle u - g(x^\ell), y \rangle$, which implies that the approximation depends on a sequence of primal points $\{x^\ell\}_{\ell=1}^k$.

2.2 Analytic Center Cutting Plane Method

As the name suggests, ACCPM calculates an analytic center of the localization set and uses the point as the next evaluation point. Let P be a polyhedron defined as follows:

$$P = \left\{ y \in \mathbb{R}^k \;\middle|\; a_\ell^T y \leq b_\ell, \, \ell = 1, \cdots, k \right\}.$$

Then, its analytic center is defined as a solution of the following unconstrained optimization problem.

$$\text{minimize} \quad -\sum_{\ell=1}^{k} \log{(b_\ell - a_\ell^T y)}.$$

The solution of the problem is not always unique. Furthermore, the problem may be infeasible. However, if P is bounded and the interior of P is nonempty, the problem has a unique solution. In ACCPM, all localization sets satisfy the condition. Therefore, we call the unique solution the analytic center.

Contrary to KCPM, ACCPM is not uniformly applicable in Step 2, because the structure of its subproblems depends on the choice of localization sets. We demonstrate the subproblems of ACCPM for

Ω^k, Ω^k_{deep}, and Ω^k_{epi}, respectively.

$$\text{minimize} \quad -\sum_{\ell=1}^{k} \log \langle g(x^\ell) - u, y - y^\ell \rangle - \sum_{j=1}^{m} \log y_j - \sum_{j=1}^{m} \log (\bar{y}_j - y_j), \tag{2.8}$$

$$\text{minimize} \quad -\sum_{\ell=1}^{k} \log (\langle g(x^\ell) - u, y - y^\ell \rangle + \min_{\ell'=1,\cdots,\ell} q_u(y^{\ell'}) - q_u(y^\ell)) - \sum_{j=1}^{m} \log y_j - \sum_{j=1}^{m} \log (\bar{y}_j - y_j),$$

$$\begin{aligned} \text{minimize} \quad & -\sum_{\ell=1}^{k} \log (r + \langle g(x^\ell) - u, y - y^\ell \rangle - q_u(y^\ell)) - \log(r - \min_{\ell=1,\cdots,k} q_u(y^\ell)) \\ & -\sum_{j=1}^{m} \log y_j - \sum_{j=1}^{m} \log (\bar{y}_j - y_j). \end{aligned} \tag{2.9}$$

Some call the version of ACCPM that use Ω^k_{epi} epigraph-ACCPM when they need to stress that it depends on the epigraph form of localization sets [5].

ACCPM has been applied for practical problems[23, 10]. Convergence properties of ACCPM have been originally studied in [1, 24]. However, they are not directly applicable to epigraph-ACCPM, because they assume that only one cutting plane is generated for an evaluation point. Convergence analysis for ACCPM with multiple cuts is given in [29].

2.3 Method of Centers of Gravity

MCG calculates the center of gravity and uses the point as the next evaluation point. MCG is not commonly used in practice, because calculating the center of gravity is generally too expensive. However, one-dimensional cases are an exception. Indeed, one-dimensional MCG is equivalent to *the bisection method*.

The bisection method is a numerical method to find a root of a continuous function. The method repeatedly chooses the middle point of the interval and selects the subinterval that contains the root. To know the method and its variation in more detail, see [19, Section 3.7]. If the dual optimization problems are univariate, every localization set boils down to an interval. Since the middle point of an interval is indeed the gravity center of the interval, it is the next evaluation point. By evaluating the dual objective and its subgradient in the middle point, we can add a cutting plane to the localization set; It is equivalent to selecting the subinterval that contains the optimal dual solution.

3 Cutting Plane Reusing Method

In cutting plane methods described in the previous section, we generate a sequence of evaluation points to solve a dual problem. Therefore, when we want to solve multiple dual problems $\{\mathcal{D}(u_t)\}_{t=1}^T$ with cutting plane methods, we need to generate multiple sequences of evaluation points. This procedure may need to be revisited in the context of multiple dual optimization problems. Since the structure of $q(y)$ is independent of u_t, we can use the common information throughout the multiple optimizations. On the other hand, only some of the evaluation points obtained for other parameters are useful.

Then, the point is how we can efficiently reuse the information. Let us consider a simple example. If there exists a common initial point, we can easily reuse it for all t. In contrast, if the optimal dual solutions are distant for each t, evaluation points on the last stage of the cutting plane methods only work for the target index t. There is almost no advantage of reusing them for other indices. However, if the optimal solutions are equal or very close for some indices, cutting planes in the late stages are likely to work well among them.

Unfortunately, we cannot know whether the optimal solutions are close or not beforehand. To overcome the difficulty, we propose an idea: We update the localization sets for other indices simultaneously as far as the new evaluation point is a member of them. We call modified cutting plane methods based on the idea by cutting plane reusing Method(CPRM)s. CPRMs generate localization sets not only for the target index t but also for other indices s as long as their localization sets include the new evaluation point. It follows that CPRMs generate a tree of evaluation points, and each evaluation point corresponds to a subset of indices. We give the procedure of CPRMs in the following.

Cutting Plane Reusing Method

Step 0. (Outer Initialization) Set $\Omega_s^0 = \{\, y \in \mathbb{R}^m \mid 0 \le y \le \bar{y} \,\}$, $S_s^0 = \{1, \cdots, T\}$, and $k_s = 0$ for all $s \in \{1, \cdots, T\}$. Set $t = 1$.

Step 1. (Inner Initialization) Set $k = k_t$.

Step 2-1. (Choosing the Next Evaluation Point) Choose $y_t^{k+1} \in \Omega_t^k$.

Step 2-2. (Filtering Companions) Set $S_t^{k+1} = \left\{\, s \in S_t^k \;\middle|\; k_s = k,\, y_t^{k+1} \in \Omega_s^k \,\right\}$.

Step 2-3. (Sharing the Next Evaluation Point) Set $y_s^{k+1} = y_t^{k+1}$ for each $s \in S_t^{k+1}$.

Step 2-4. (Sharing Companions) Set $S_s^{k+1} = S_t^{k+1}$ for each $s \in S_t^{k+1}$.

Step 2-5. (Updating the Frontier) Update $k_s = k_s + 1$ for each $s \in S_t^{k+1}$.

Step 3. (Calling an Oracle) Obtain $x_t^{k+1} \in \operatorname{argmin}\left\{\, f(x) + \langle y_t^{k+1}, g(x) \rangle \;\middle|\; x \in X \,\right\}$.

Step 4. (Updating the Localization Sets) Update $\Omega_s^{k+1} = \Omega_s^k \cap \left\{\, y \in \mathbb{R}^m \;\middle|\; \langle u_s - g(x_t^{k+1}), y - y_t^{k+1} \rangle \le 0 \,\right\}$ for each $s \in S_t^{k+1}$.

Step 5. (Inner Stopping Criteria) If Ω_t^{k+1} is sufficiently small, go to Step 7.

Step 6. (Updaing the Inner Iteration) Update $k = k + 1$, then go to Step 2-1.

Step 7. (Updating the Outer Iteration) If $t = T$, exit. Otherwise, set $t = t + 1$ and go to Step 1.

A CPRM consists of outer and inner loops. Outer loops correspond to updating the indices $t = 1, \cdots, T$, and inners loops correspond to a cutting plane method for each u_t. However, not only the localization set of the target index Ω_t^k, but also localization sets of other indices may be updated in Step 4. The set of indices of which localization sets are simultaneously updated is determined in Step 2-2. We call the indices *companions* and denote the set of k-th companions of t by S_t^k. Note that the index t itself is always a member of S_t^k.

The algorithm generates multiple sequences of evaluation points. The multiple sequences can be compiled into a tree of evaluation points because evaluation points are shared among companions. Let us denote the k-th evaluation point of index t by y_t^k. To emphasize that y_t^k is shared among $s \in S_t^k$, we denote y_t^k by $y_{S_t^k}^k$. For example, if y_1^2 is shared among $s = 1, 3, 6$, i.e, $S_1^2 = S_3^2 = S_6^2 = \{1, 3, 6\}$, we have $y_{\{1,3,6\}}^2 = y_1^2 = y_3^2 = y_6^2$. One may think that an index s can be included in more than two different sets of companions in the same degree k simultaneously, which implies that the evaluation point y_s^k cannot be uniquely determined. However, such a case does not occur due to the first condition $k = k_s$ in Step 2-2, the role of which may not be evident at first glance. As the inner iteration proceeds, the set of companions shrinks and usually becomes a singleton.

When we go to the next inner loop, we have already obtained a sequence of evaluation points. Therefore, we can start the procedure from a small localization set. To inherit the information of past evaluation points and companions, Steps 2-3 to 2-5 are necessary. Convergence properties of a CPRM depend on those of the cutting plane method that the CPRM adapts in the inner loops.

To understand the structure of CPRMs in more detail, we demonstrate two small examples. The first one is a group of knapsack problems. We solve the problems with a CPRM based on MCG. The second one is a group of two-dimensional knapsack problems that is solved by a CPRM based on KCPM. We use bold style in the examples, such as $\boldsymbol{x}$ and $\boldsymbol{u}$, when we need to stress that it is a vector.

Example 3.1. *Three Knapsack problems*

$$\begin{array}{rl} \text{minimize} & -x_1 - 2x_2 - 3x_3 - 4x_4 \\ \text{subject to} & 2x_1 + 2x_2 + 2x_3 + 2x_4 \leq u, \\ & x_i \in \{0,1\},\ i = 1, \cdots, 4, \end{array}$$

where the parameter u is variable among $\{u_t\}_{t=1}^3 = \{7,5,3\}$.

We can easily confirm that the dual optimal solution for each $t = 1,2,3$ is $\frac{1}{2}, 1, \frac{3}{2}$, respectively. However, to understand the behaviour of CPRMs, let us check the procedure in detail. The CPRM used in this example is based on MCG. We also assume $\bar{y} = 5$.

Step 0. Set $\Omega_1^0 = \Omega_2^0 = \Omega_3^0 = [0,5]$, $S_1^0 = S_2^0 = S_3^0 = \{1,2,3\}$, $k_1 = k_2 = k_3 = 0$, and $t = 1$.

Step 1. Set $k = k_1 = 0$.

Step 2-1. Bisection method returns $y_1^1 = \frac{5}{2} \in [0,5] = \Omega_1^0$.

Step 2-2. Since $S_1^0 = \{1,2,3\}$, $k_1 = k_2 = k_3 = 0$, and $y_1^1 \in \Omega_1^0 = \Omega_2^0 = \Omega_3^0$, we set $S_1^1 = \{1,2,3\}$.

Step 2-3 to 2-5. Set $y_2^1 = y_3^1 = y_1^1 = \frac{5}{2}$ and $S_2^1 = S_3^1 = S_1^1 = \{1,2,3\}$. Update $k_1 = k_2 = k_3 = 0+1 = 1$.

Step 3. Since $f(\boldsymbol{x}) + \langle y_1^1, g(\boldsymbol{x})\rangle = 4x_1 + 3x_2 + 2x_3 + x_4$, we have $\boldsymbol{x}_1^1 = (0,0,0,0)$.

Step 4. Since $g(\boldsymbol{x}_1^1) = 0$, we have $\Omega_1^1 = \Omega_2^1 = \Omega_3^1 = [0, \frac{5}{2}]$.

Step 5, 6 Update $k = 1$ and go to Step 2-1.

The first inner loop is straightforward because the first evaluation point is common for all indices. We also confirm that the effect of cutting planes is equal in Step 4, which is not true in the second inner loop. The inner loop between Step 2-1 to Step 6 is indeed the bisection method except for updating y_s^k and Ω_s^k for other indices s.

Step 2-1. Bisection method returns $y_1^2 = \frac{5}{4} \in [0, \frac{5}{2}] = \Omega_1^1$.

Step 2-2. Since $S_1^1 = \{1,2,3\}$, $k_1 = k_2 = k_3 = 1$, and $y_1^2 \in \Omega_1^1 = \Omega_2^1 = \Omega_3^1$, we set $S_1^2 = \{1,2,3\}$.

Step 2-3 to 2-5. Set $y_2^2 = y_3^2 = y_1^2 = \frac{5}{4}$ and $S_2^2 = S_3^2 = S_1^2 = \{1,2,3\}$. Update $k_1 = k_2 = k_3 = 1+1 = 2$.

Step 3. Since $f(\boldsymbol{x}) + \langle y_1^2, g(\boldsymbol{x})\rangle = \frac{3}{2}x_1 + \frac{1}{2}x_2 - \frac{1}{2}x_3 - \frac{3}{2}x_4$, we have $\boldsymbol{x}_1^2 = (0,0,1,1)$.

Step 4. Since $g(\boldsymbol{x}_1^2) = 4$, we have $\Omega_1^2 = \Omega_2^2 = [0, \frac{5}{4}]$ and $\Omega_3^2 = [\frac{5}{4}, \frac{5}{2}]$.

Step 5, 6. Update $k = 2$ and go to Step 2-1.

Similar to the first inner loop, the evaluation point $y_1^2 = \frac{5}{4}$ is commonly used for all indices. However, the effects of the cutting planes differ in Step 4, because the signs of $u_s - g(x_1^2)$ are different among $s \in S_1^2$.

Step 2-1. Bisection method returns $y_1^3 = \frac{5}{8} \in [0, \frac{5}{4}] = \Omega_1^2$.

Step 2-2. Since $S_1^2 = \{1,2,3\}$, $k_1 = k_2 = k_3 = 2$, $y_1^3 \in \Omega_1^2 = \Omega_2^2 = [0, \frac{5}{4}]$, and $y_1^3 \notin \Omega_3^2 = [\frac{5}{4}, \frac{5}{2}]$, we set $S_1^3 = \{1,2\}$.

Step 2-3 to 2-5. Set $y_2^3 = y_1^3 = \frac{5}{8}$ and $S_2^3 = S_1^3 = \{1,2\}$. Update $k_1 = k_2 = 2+1 = 3$.

Step 3. Since $f(\boldsymbol{x}) + \langle y_1^3, g(\boldsymbol{x})\rangle = \frac{1}{4}x_1 - \frac{3}{4}x_2 - \frac{7}{4}x_3 - \frac{11}{4}x_4$, we have $\boldsymbol{x}_1^3 = (0,1,1,1)$.

Step 4. Since $g(\boldsymbol{x}_1^3) = 6$, we have $\Omega_1^3 = [0, \frac{5}{8}]$ and $\Omega_2^3 = [\frac{5}{8}, \frac{5}{4}]$.

Step 5, 6. Update $k = 3$ and go to Step 2-1.

In the third inner loop, the evaluation point $y_1^3 = \frac{5}{8}$ is used only for $s = 1, 2$, because the localization sets are different among S_1^2. The effects of cutting planes also differ in Step 4, which affects the filtering in Step 2-2 in the next loop.

Step 2-1. Bisection method returns $y_1^4 = \frac{5}{16} \in [0, \frac{5}{8}] = \Omega_1^3$.

Step 2-2. Since $S_1^3 = \{1, 2\}$, $k_1 = k_2 = 3$, $y_1^4 \in \Omega_1^3 = [0, \frac{5}{8}]$, and $y_1^4 \notin \Omega_2^3 = [\frac{5}{8}, \frac{5}{4}]$, we set $S_1^4 = \{1\}$.

Step 2-3 to 2-5. Update $k_1 = 3 + 1 = 4$.

Step 3. Since $f(\boldsymbol{x}) + \langle y_1^4, g(\boldsymbol{x})\rangle = -\frac{3}{8}x_1 - \frac{11}{8}x_2 - \frac{19}{8}x_3 - \frac{27}{8}x_4$, we have $\boldsymbol{x}_1^4 = (1, 1, 1, 1)$.

Step 4. Since $g(\boldsymbol{x}_1^4) = 8$, we have $\Omega_1^4 = [\frac{5}{16}, \frac{5}{8}]$.

Step 5, 6. Update $k = 4$ and go to Step 2-1.

In the fourth inner loop, the set of companions consists of only the target index $t = 1$ after filtering. Therefore, we need not explain the procedure after the fifth degree, because the inner loops become just a bisection method. The inner loop continues until the criteria in Step 5 is satisfied. Then, we go to Step 7, update $t = 2$, and go to Step 1. Before going to the second outer iteration, we demonstrate the tree of evaluation points obtained in the first outer iteration in Figure 2.

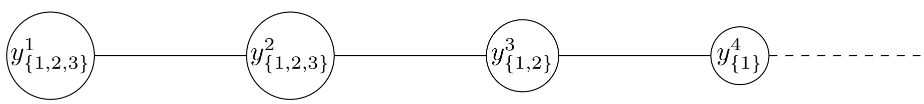

Figure 2: The tree of evaluation points after the first outer iteration

Step 1. Set $k = k_2 = 3$.

Step 2-1. Bisection method returns $y_2^4 = \frac{15}{16} \in [\frac{5}{8}, \frac{5}{4}] = \Omega_2^3$.

Step 2-2. Since $S_2^3 = \{1, 2\}$, $k_1 > k = k_2 = 3$, and $y_2^4 \in \Omega_2^3$, we set $S_2^4 = \{2\}$.

Step 2-3 to 2-5. Update $k_2 = 3 + 1 = 4$.

Step 3. Since $f(\boldsymbol{x}) + \langle y_2^4, g(\boldsymbol{x})\rangle = \frac{7}{8}x_1 - \frac{1}{8}x_2 - \frac{9}{8}x_3 - \frac{17}{8}x_4$, we have $\boldsymbol{x}_2^4 = (0, 1, 1, 1)$.

Step 4. Since $g(\boldsymbol{x}_2^4) = 6$, we have $\Omega_2^4 = [\frac{15}{16}, \frac{5}{4}]$.

Step 5, 6. Update $k = 4$ and go to Step 2-1.

In the first inner loop of the second outer loop, we need to notice that index $s = 1$ cannot be filtered in Step 2-1, because degree 3 is not the frontier of index 1. After Step 2-1, the set of companions consists of only the target index $t = 2$. Therefore, we need not explain the rest of the inner loops. The inner loops continue until the criteria in Step 5 are satisfied. Then, we go to Step 7, update $t = 3$, and go to Step 1. We illustrate the tree of evaluation points obtained in the second outer iteration in Figure 3.

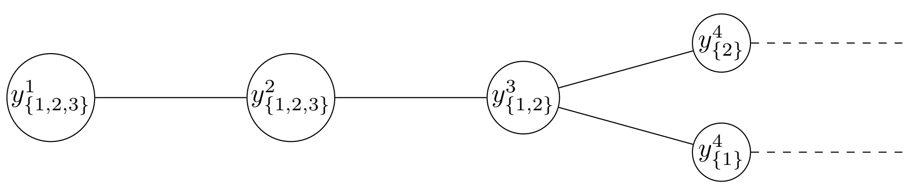

Figure 3: The tree of evaluation points after the second outer iteration

Step 1. Set $k = k_3 = 2$.

Step 2-1. Bisection method returns $y_3^3 = \frac{15}{8} \in [\frac{5}{4}, \frac{5}{2}] = \Omega_2^3$.

Step 2-2. Since $S_3^2 = \{1, 2, 3\}$, $k_1 > k = 2$, $k_2 > k = 2$, $k_3 = k = 2$, and $y_3^3 \in \Omega_2^3$, we set $S_3^3 = \{3\}$.

Step 2-3 to 2-5. Update $k_3 = 2 + 1 = 3$.

Step 3. Since $f(\boldsymbol{x}) + \langle y_3^3, g(\boldsymbol{x})\rangle = \frac{11}{4}x_1 + \frac{7}{4}x_2 + \frac{3}{4}x_3 - \frac{1}{4}x_4$, we have $\boldsymbol{x}_2^4 = (0, 0, 0, 1)$.

Step 4. Since $g(\boldsymbol{x}_2^4) = 2$, we have $\Omega_3^3 = [\frac{5}{4}, \frac{15}{8}]$.

Step 5, 6. Update $k = 3$ and go to Step 2-1.

After Step 2-1, the set of companions becomes a singleton, which implies that the rest of the inner loops become the bisection method. The inner loops continue until the criteria in Step 5-1 are satisfied, and we finish the algorithm in Step 7. Finally, we obtain the tree of the evaluation points in Figure 4. Figure

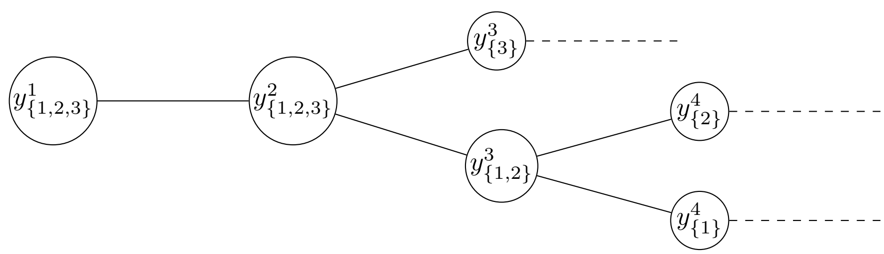

Figure 4: The tree of evaluation points after the outer final iteration

5 visualizes the relationships among the evaluation points.

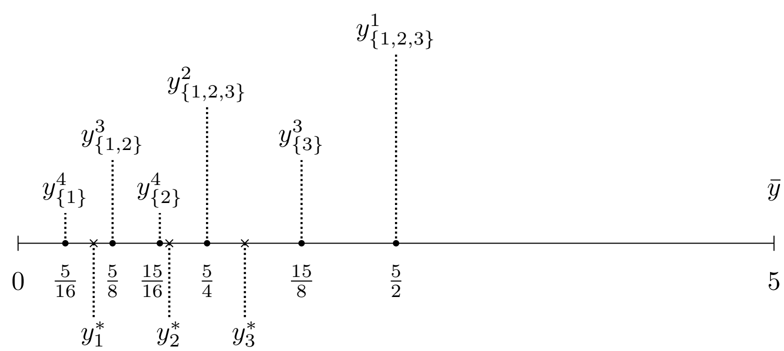

Figure 5: Evaluation points in the number line

We stress that the behaviour of CPRMs is particular in one-dimensional problems in the sense that the localization sets are equal among companions. This property does not hold for multi-dimensional problems. We explain it later in this section. We use a CPRM based on KCPM in the following example. We also assume $\bar{\boldsymbol{y}} = (5, 5)$.

Example 3.2. *Four 2-dimensional Knapsack problems*

$$\begin{aligned} \text{minimize} \quad & -x_1 - 2x_2 - 3x_3 - 4x_4 \\ \text{subject to} \quad & 2x_1 + 2x_2 + 2x_3 + 2x_4 \le u_1, \\ & 3x_3 + 3x_4 \le u_2, \\ & x_i \in \{0, 1\},\ i = 1, \cdots, 4, \end{aligned}$$

where the parameter $\boldsymbol{u} = (u_1, u_2)$ *is variable among* $\{\boldsymbol{u}_t\}_{t=1}^4 = \{(1, 4), (5, 2), (3, 5), (7, 4)\}$.

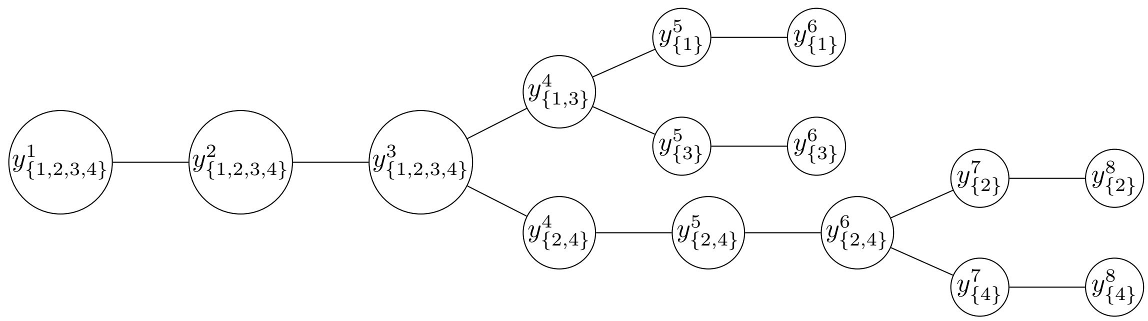

Figure 6: The final tree of evaluation points.

Demonstrating the detailed procedure of CPRMs for multi-dimensional problems is so complicated that we send the body to the appendix. We only display the final tree of evaluation points in Figure 6. However, the appendix is helpful for readers eager to implement a CPRM.

Let us consider the difference between one-dimensional problems and multi-dimensional problems. Even if two indices are companions in multi-dimensional problems, their localization sets must differ. Furthermore, their localization sets may intersect even if the two indices are not companions. For example, we illustrate the second localization sets for indices $t = 1$ and 2, in Figure 7. They are different despite the two indices sharing the same evaluation points until $k = 3$. We also illustrate the third localization sets for the two indices in Figure 8. Although the two indices have different evaluation points in $k = 4$, their localization sets are not separated.

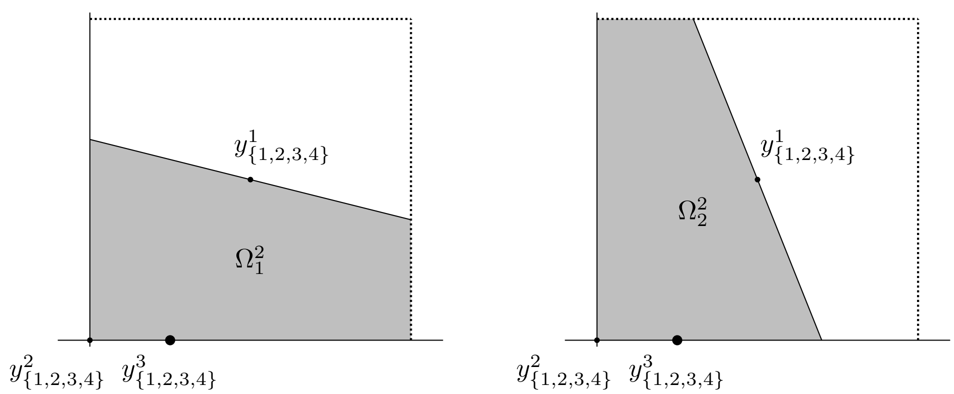

Figure 7: Comparison of Localization Sets in degree 2.

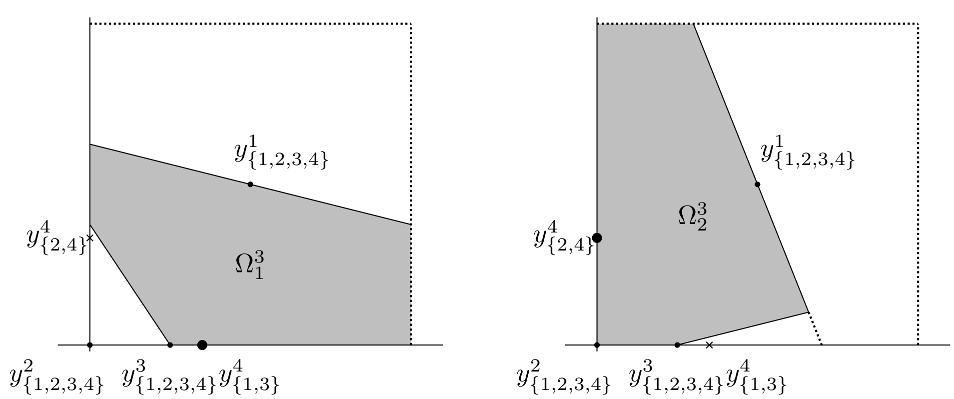

Figure 8: Comparison of Localization sets in degree 3.

4 Numerical Experiments

In this section, we demonstrate some numerical experiments to confirm the efficiency of CPRMs. We prepare two types of problems. The first one is shortest path problems with resource constraints(SPPRC), and the second one is unit commitment problems(UCP). As these two problems are suitable for Lagrangian relaxation, evaluations of the dual functions are expensive.

In our numerical experiments, main parts of the algorithms are implemented by the author with Julia 1.6 on windows 10 OS. We execute the optimization problems on a computer with Intel Core i5-8265 1.6GHz CPU.

We stress that our goal in the numerical experiments is not demonstrating the superiority of the implementation over other sophisticated ones tailored for each problem but demonstrating the advantage of CPRMs from the viewpoint of reducing evaluations.

4.1 Methods and Software

To execute a CPRM, we need to choose a cutting plane method. In our numerical experiments, we have chosen KCPM and ACCPM. We abbreviate a CPRM based on KCPM to KCPRM: Kelley's cutting plane reusing method. Similarly, we abbreviate a CPRM based on ACCPM to ACCPRM: Analytic center cutting plane reusing method.

To evaluate the performance of KCPRM, we compare it with KCPM. Since our goal is solving a group of dual problems $\{\mathcal{D}(u_t)\}_{t=1}^{T}$, KCPM is independently applied to each problem in the group in the latter case. In addition to the two methods, we prepare warm start KCPM. In warm start KCPM, we use an optimal solution of the problems as an initial point of the next problem. Other parts of warm start KCPM are equal to KCPM. We call the method KCPM-ws. We compare the performance among KCPM, KCPRM, and KCPM-ws. Similarly, we compare ACCPM, ACCPRM, and ACCPM-ws. We do not aim to compare KCPRM and ACCPRM because comparison among cutting plane methods is not the scope of the paper.

In applying these methods, the subproblem of KCPM is solved by CLP [9] through JuMP [8]. On the other hand, ACCPM is implemented by the author. We do not insist that our choice is one of the most sophisticated. However, our aim is not to pursue best performance of cutting plane methods but to confirm the advantages of CPRMs from the viewpoint of reducing evaluations. Our choice is sufficient enough to confirm them.

4.2 Stopping Criteria

In Section 2, we have demonstrated that the cutting plane methods terminate if the localization set is small enough. However, estimating the volume of a localization set is too expensive. Therefore, we need other criteria for us to use the methods in practice. Defining a uniform stopping criterion among cutting plane methods is not a easy task. For example, the distance between the next evaluation point and the bound of the localization set is proposed in [1] as a stopping criterion for ACCPM. Although it is sutitable for ACCPM, it does not work well for KCPM; Even though the next evaluation point is on the bound of the localization set, it does not mean that the set is small enough in KCPM. For example, see Figure 7.

To measuer the accuracy of the solution, following value is often used in the context of nonsmooth convex optimization [15, 16, 27].

$$\epsilon = \frac{q_u^{best} - q_u^*}{1 + |q_u^*|}, \tag{4.1}$$

where q_u^{best} is the minimum value of the objective obtained so far, and q_u^* is the optimal value of the problem $\mathcal{D}(u)$. Unfortunately, the optimal value q_u^* is usually unknown beforehand. Therefore, we need an alternative to q_u^* for each cutting plane method. In KCPM, we use the following value as a substitute.

$$\epsilon_{\text{kcpm}} = \frac{q_u^{best} - L_u}{1 + |L_u|}, \tag{4.2}$$

where L_u is the optimal value of the k-th subproblem of KCPM:

$$L_u = \min_{0 \le y \le \bar{y}} \max_{\ell=1,\cdots,k} (q_u(y) + \langle u - g(x^\ell), y - y^\ell \rangle). \tag{4.3}$$

A disadvantage of $\epsilon_{\rm kcpm}$ is that calculating L_u is essentially equivalent to solving a subproblem of KCPM. Therefore, it is too expensive to use for other cutting plane methods. In ACCPM, we use another value:

$$\epsilon_{\rm accpm} = \frac{q_u^{best} - \hat{L}_u}{1 + |\hat{L}_u|}, \tag{4.4}$$

where $\hat{L}_u$ is defined for Ω^k as follows:

$$\hat{L}_u = -\frac{1}{\langle \mathbf{1}, \tau \rangle} \left\{ \sum_{\ell=1}^{k} \tau_\ell f(x^\ell) + \sum_{j=1}^{m} \frac{\bar{y}_j - y_j^\ell}{\bar{y}_j} \right\}, \tag{4.5}$$

$$\tau_\ell = -\frac{1}{\langle u - g(x^\ell), y^{k+1} - y^\ell \rangle}, \tag{4.6}$$

where y^{k+1} is a next evaluation point, i.e, the analytic center of Ω^k. $\hat{L}_u$ has two advantages: It can be easily caluculated as a byproduct of ACCPM and it is lower than L_u.

Lemma 4.1. *For a localization set Ω^k defined by a sequence of evaluation points $\{y_\ell\}_{\ell=1}^k$, let L_u and $\hat{L}_u$ be parameters defined by (4.3) and (4.5), respectively. Then, following relation holds for all u such that $\mathcal{D}(u)$ has an optimal solution.*

$$\hat{L}_u \le L_u. \tag{4.7}$$

Since the proof of Lemma 4.1 is not immediate [5, Section 4], we send it to the Appendix. The inequality directly proves that

$$\epsilon \le \epsilon_{\rm kcpm} \le \epsilon_{\rm accpm}. \tag{4.8}$$

In our numerical expriments, we use the termination critera $\epsilon_{\rm kcpm} \le 10^{-5}$ and $\epsilon_{\rm accpm} \le 10^{-5}$ for KCPM and ACCPM, respectively. In view of (4.8), both gurantee $\epsilon \le 10^{-5}$.

4.3 Shortest Path Problems with Resource Constraints

As the name suggests, SPPRC is a shortest path problem equipped with some resource constraints. We give its formulation as follows.

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in E} c_{ij} x_{ij} && (4.9)\\
\text{subject to} \quad & \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \delta_{1i} - \delta_{iN},\ i \in \{1, \cdots, N\}, && (4.10)\\
& \sum_{(i,j)\in E} a_{ijr} x_{ij} \le u_r,\ r \in \{1, \cdots, R\}, && (4.11)\\
& x_{ij} \in \{0, 1\},\ (i,j) \in E, && (4.12)
\end{aligned}$$

where E be a set of edges, N be a number of nodes, R be a number of resources, c_{ij} be a cost to use edge (i,j), u_r be a resource bound for resource r, a_{ijr} be a necessary resource r required to use an edge (i,j), and x_{ij} be a binary variable whether to use edge (i,j) or not.

Although Lagrangian relaxation is a reliable approach for solving the problem, we need to solve a shortest path problem to evaluate the dual function. Therefore, reducing the total number of evaluations is essential.

In the numerical experiments, we prepare 5 groups of randomly generated problems in which the number of the nodes are $1000, 2000, 4000, 8000$, and 16000, the number of the edges are 10 times as large as that of nodes, and the number of resources is 5. Each problem in a group has same core structure except for the resource bounds. Parametric resource bounds are generated in the following. First, we solve the shortest path problem without resource constraints, and the solution provides us necessary resources for the shortest path. Next, we multiply the necessary resources by 0.8 or 0.9 for each resource. We prepare 7 patterns of multiplication: (0.8, 0.8, 0.8, 0.8, 0.8), (0.9, 0.8, 0.8, 0.8, 0.8), (0.8, 0.9, 0.8, 0.8, 0.8), (0.8, 0.8, 0.9, 0.8, 0.8),(0.8, 0.8, 0.8, 0.9, 0.8), (0.8, 0.8, 0.8, 0.8, 0.9), (0.9, 0.9, 0.9, 0.9, 0.9). Therefore, each group of problems consists of 7 problems. Other features of the problems are generated by the rule written in [2].

		Number of total evaluations					
N	T	KCPM	KCPRM	KCPM-ws	ACCPM	ACCPRM	ACCPM-ws
1000	7	82	26	67	478	160	374
2000	7	109	30	98	450	227	418
4000	7	83	26	87	413	275	359
8000	7	78	21	75	408	141	307
16000	7	66	26	58	321	178	306

Table 1: Number of total evaluations for SPPRC

Table 1 compares the number of total evaluations for the 5 groups of problems. KCPRM dramatically outperforms KCPM, and KCPM-ws also works to reduce the evaluations. However, the effects are much less than KCPRM. We observe a similar tendency among ACCPM, ACCPRM, and ACCPM-ws.

		Calculation times(sec)					
N	T	KCPM	KCPRM	KCPM-ws	ACCPM	ACCPRM	ACCPM-ws
1000	7	15.8	14.8	15.5	15.6	7.2	12.4
2000	7	20.2	16.3	19.5	28.3	16.0	26.5
4000	7	43.5	24.6	44.8	157.5	99.5	134.4
8000	7	103.6	45.6	105.0	383.4	156.2	274.2
16000	7	413.4	198.2	360.9	1949.3	1074.4	1820.8

Table 2: Calculation times for SPPRC

We compare calculation times for the same groups of problems in Table 2. If the size of the problems are small, differences of the calculation time among KCPM, KCPRM, and KCPM-ws are not as large as those of the number of evaluations. It suggests that solving subproblems are more dominant than evaluations in small problems. However, in both KCPMs and ACCPMs, the cost of solving subproblems become relatively negligible as the problems become larger.

4.4 Unit Commitment Problems

Under the electricity demand, UCP aims to optimize the operation of electrical power generating units. Since the electrical power system is one of the society's most critical system, UCPs have long history and many variants. To know them in detail, see [26, 14] and the references therein. In this numerical experiment, we consider a variation of UCP that assumes the following [17]:

(U1) Electricity demand is deterministic.

(U2) Power generating units have start-up costs.

(U3) Power generating units must obey minimum-up and minimum-down requirements.

(U4) Power generating units have quadratic cost functions.

The mathematical formulation of the problem is

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J}\left(s_i v_{ij} + a_i z_{ij} + b_i x_{ij} + c_i x_{ij}^2\right), \\
\text{subject to} \quad & \sum_{i=1}^{I} x_{ij} \geq u_j,\ j = 1, \dots, J, \\
& \sum_{\tau=j-M_i^{up}}^{j} v_{i\tau} \leq z_{ij},\ i = 1, \dots, I, \\
& \sum_{\tau=j-M_i^{down}}^{j} w_{i\tau} \leq 1 - z_{ij},\ i = 1, \dots, I, \\
& \underline{p}_i z_{ij} \leq x_{ij} \leq \bar{p}_i z_{ij},\ i = 1, \dots, I,\ j = 1, \dots, J, \\
& z_{ij} \in \{0,\, 1\}, v_{ij} \in \{0,\, 1\}, w_{ij} \in \{0,\, 1\},
\end{aligned}
\tag{4.13}
$$

where I stands for the number of units and J is the length of the period, z_{ij} means whether i-th unit is active or not at time j, x_{ij} stands for the electricity that i-th unit generates at time j, $v_{i\tau}$ and $w_{i\tau}$ are binary variables that mean whether i-th unit is switched on or not at time τ and whether i-th unit is switched off or not at time τ respectively, s_i stands for the start-up cost of i-th unit, a_i stands for the fixed cost of i-th unit, b_i and c_i are first and second order cost coefficients of i-th unit, respectively, u_j stands for electricity demands at time j, $\underline{p}_i$ and $\bar{p}_i$ are minimum and maximum output of i-th unit, respectively, M_i^{up} and M_i^{down} stands for the duration of minimum-up and minimum-down of i-th unit, respectively. To obtain its dual problem, the demand constraints (4.13) are relaxed with dual variables $y \in \mathbb{R}_+^J$. We can efficiently solve the Lagrangian relaxation problem for a given y by dynamic programming for each unit j.

In the numerical experiments, we prepare 3 groups of randomly generated problems in which the number of the units are $50, 500$, and 5000, and the length of the period is 24. First 5 units are Unit 1 to 5 described in [17, Table 1]. The remaining units are randomly generated from the 5 units to mimic them. All problems in a group have the same core structure except for the demand vector u. Parametric demand vectors are generated in the following. First, we prepare a basic demand vector for each group. To obtain the basic demand vectors, we multiply a demand vector given in [17, Table 1] by $10, 100$, and 1000 for $I = 50, 500$, and 5000, respectively. Next, we multiply the basic demand vector by 1.0 or 1.1 for each element. We prepare 25 patterns of multiplication: (1.0, 1.0, ..., 1.0), (1.1, 1.0, ..., 1.0), (1.0, 1.1, ..., 1.0), ..., (1.0, 1.0, 1.0, ..., 1.1). Therefore, each group consists of 25 problems.

		Number of total evaluations					
I	T	KCPM	KCPRM	KCPM-ws	ACCPM	ACCPRM	ACCPM-ws
50	25	5870	2960	5781	4312	2776	3934
500	25	8657	4379	8628	4714	3671	4520
5000	25	8760	4783	8696	4643	3672	4646

Table 3: Number of total evaluations for UCP

		Calculation times(sec)					
I	T	KCPM	KCPRM	KCPM-ws	ACCPM	ACCPRM	ACCPM-ws
50	25	20.8	33.7	21.0	31.2	40.9	32.8
500	25	86.4	78.6	85.0	68.2	71.0	63.3
5000	25	589.5	388.8	580.6	307.2	246.0	317.4

Table 4: Calculation times for UCP

Table 3 compare the number of total evaluations for the 3 groups of problems. As observed in SPPRC, KCPRM and ACCPRM remarkably outperform KCPM and ACCPM and are more effective than KCPM-ws and ACCPM-ws, respectively, in reducing the number of evaluations. On the other hand, when it comes to total calculation time, table 4 suggests that KCPRM and ACCPRM are not best for small problems. In small problems, solving subproblems is more dominant than evaluations. In addition, table 4 implies that solving a subproblem in KCPRM is more expensive than that of KCPM in $I = 50$. The size of subproblems causes this mysterious phenomenon. The size of subproblems in KCPRM often become large even in the early stage of the inner iterations if they reuse lots of past cutting planes. Therefore, the average calculation cost for solving a subproblem in KCPRM is often larger than that of KCPM. The same problem holds between ACCPM and ACCPRM. It follows that a CPRM does not always outperform its corresponding cutting plane method. For large problems, KCPRM and ACCPRM outperform KCPM and ACCPM, respectively.

5 Conclusion

This paper aims to solve multiple dual optimization problems efficiently. The aim is different from solving a multi-objective optimization problem or a parametric optimization problem.

To solve the problems efficiently, we have proposed cutting plane reusing methods. Since the proposed methods efficiently reuse the past information of cutting planes, they dramatically reduce the total number of evaluations. Although the methods reuse past information, they are different from warm-start. We demonstrated the structure of the methods in detail through two concrete examples. We also confirmed the advantage of the proposed methods through two numerical experiments.

Solving multiple optimization problems has yet to be widely researched as far as we know. In general, we solve each problem independently. However, simultaneously solving multiple problems is sometimes efficient if they share a core structure. For example, when we need to solve all path problems in the same network, applying the Floyd-Warshall algorithm once is better than multiple applications of Dijkstr's algorithm.

For solving multiple dual optimization problems, we have focused on cutting plane methods and developed a framework based on them. However, some other methods are also promising. Pursuing the possibilities of other methods are subject of further research. We hope other modified methods tailored for multiple optimizations will be proposed and confirmed to be efficient.

References

[1] David S Atkinson and Pravin M Vaidya. A cutting plane algorithm for convex programming that uses analytic centers. *Mathematical Programming*, 69(1):1–43, 1995.

[2] John E Beasley and Nicos Christofides. An algorithm for the resource constrained shortest path problem. *Networks*, 19(4):379–394, 1989.

[3] Joseph-Frédéric Bonnans, Jean Charles Gilbert, Claude Lemaréchal, and Claudia Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer Science & Business Media, 2006.

[4] Stephen Boyd and Lieven Vandenberghe. Localization and cutting-plane methods. *From Stanford EE 364b lecture notes*, 2007.

[5] Stephen Boyd, Lieven Vandenberghe, and Joëlle Skaf. Analytic center cutting-plane method. *Lecture Notes from Stanford University*, 2008.

[6] Yann Collette and Patrick Siarry. *Multiobjective optimization: principles and case studies*. Springer Science & Business Media, 2004.

[7] Guy Desaulniers, Jacques Desrosiers, and Marius M Solomon. *Column generation*, volume 5. Springer Science & Business Media, 2006.

[8] Iain Dunning, Joey Huchette, and Miles Lubin. Jump: A modeling language for mathematical optimization. *SIAM review*, 59(2):295–320, 2017.

[9] JJ Forrest, D dl Nuez, and R Lougee-Heimer. Clp: Coin-or linear programming solver (2012).

[10] J-L Goffin, Jacek Gondzio, Robert Sarkissian, and J-P Vial. Solving nonlinear multicommodity flow problems by the analytic center cutting plane method. *Mathematical programming*, 76(1):131–154, 1997.

[11] Jean-Louis Goffin, Alain Haurie, and Jean-Philippe Vial. Decomposition and nondifferentiable optimization with the projective algorithm. *Management science*, 38(2):284–302, 1992.

[12] Jean-Louis Goffin and Jean-Philippe Vial. Convex nondifferentiable optimization: A survey focused on the analytic center cutting plane method. *Optimization methods and software*, 17(5):805–867, 2002.

[13] JB Hiriart-Urruty and Claude Lemaréchal. *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*. Springer-Verlag, New York, 1993.

[14] Yuping Huang, Panos M Pardalos, and Qipeng P Zheng. *Electrical power unit commitment: deterministic and two-stage stochastic programming models and algorithms*. Springer, 2017.

[15] Napsu Karmitsa, Adil Bagirov, and Marko M Mäkelä. Empirical and theoretical comparisons of several nonsmooth minimization methods and software. Technical report, TUCS Technical Report 959, Turku Centre for Computer Science, 2009.

[16] Napsu Karmitsa and Marko M Mäkelä. Limited memory bundle method for large bound constrained nonsmooth optimization: convergence analysis. *Optimization Methods & Software*, 25(6):895–916, 2010.

[17] Spyros A Kazarlis, AG Bakirtzis, and Vassilios Petridis. A genetic algorithm solution to the unit commitment problem. *IEEE transactions on power systems*, 11(1):83–92, 1996.

[18] James E Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[19] Mykel J Kochenderfer and Tim A Wheeler. *Algorithms for optimization*. Mit Press, 2019.

[20] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. *arXiv preprint arXiv:1508.04874*, 2015.

[21] Claude Lemaréchal. Chapter vii nondifferentiable optimization. *Handbooks in operations research and management science*, 1:529–572, 1989.

[22] Anatoly Yur'evich Levin. An algorithm for minimizing convex functions. In *Doklady Akademii Nauk*, volume 160, pages 1244–1247. Russian Academy of Sciences, 1965.

[23] O du Merle, J-L Goffin, C Trouiller, and J-P Vial. A lagrangian relaxation of the capacitated multi-item lot sizing problem solved with an interior point cutting plane algorithm. In *Approximation and Complexity in Numerical Optimization*, pages 380–405. Springer, 2000.

[24] Yu Nesterov. Complexity estimates of some cutting plane methods based on the analytic barrier. *Mathematical Programming*, 69(1):149–176, 1995.

[25] Yurii Nesterov. Nonsmooth convex optimization. In *Lectures on Convex Optimization*, pages 139–240. Springer, 2018.

[26] Narayana Prasad Padhy. Unit commitment-a bibliographical survey. *IEEE Transactions on power systems*, 19(2):1196–1205, 2004.

[27] Claudia Sagastizábal. Composite proximal bundle method. *Mathematical Programming*, 140(1):189–233, 2013.

[28] Georg Still. Lectures on parametric optimization: An introduction. *Optimization Online*, 2018.

[29] Yinyu Ye. Complexity analysis of the analytic center cutting plane method that uses multiple cuts. *Mathematical Programming*, 78(1):85–104, 1996.

[30] Lofti Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE transactions on Automatic Control*, 8(1):59–60, 1963.

A Appendix

A.1 Procedure of CPRM for Example 3.2

In this subsection, we demonstrate the detailed procedure of a CPRM applied to Example 3.2. The CPRM is based on KCPM.

Step 0. Set $\Omega_1^0 = \Omega_2^0 = \Omega_3^0 = \Omega_4^0 = [0,5]^2$, $S_1^0 = S_2^0 = S_3^0 = S_4^0 = \{1,2,3,4\}$, $k_1 = k_2 = k_3 = k_4 = 0$, and $t = 1$.

Step 1. Set $k = k_1 = 0$.

Step 2-1. Initial point $\boldsymbol{y}_1^1 = (\frac{5}{2}, \frac{5}{2})$ is a center of $\Omega_1^0 = [0,5]^2$.

Step 2-2. Since $S_1^0 = \{1,2,3,4\}$, $k_1 = k_2 = k_3 = k_4 = 0$, and $\boldsymbol{y}_1^1 \in \Omega_1^0 = \Omega_2^0 = \Omega_3^0 = \Omega_4^0$, we set $S_1^1 = \{1,2,3,4\}$.

Step 2-3 to 2-5. Set $\boldsymbol{y}_2^1 = \boldsymbol{y}_3^1 = \boldsymbol{y}_4^1 = \boldsymbol{y}_1^1 = (\frac{5}{2}, \frac{5}{2})$ and $S_2^1 = S_3^1 = S_4^1 = S_1^1 = \{1,2,3,4\}$. Update $k_1 = k_2 = k_3 = k_4 = 0 + 1 = 1$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_1^1, g(\boldsymbol{x}) \rangle = 4x_1 + 3x_2 + \frac{19}{2}x_3 + \frac{17}{2}x_4$, we have $\boldsymbol{x}_1^1 = (0,0,0,0)$.

Step 4. Since $g(\boldsymbol{x}_1^1) = (0,0)$, we have

$$\Omega_1^1 = [0,5]^2 \cap \left\{ (y_1, y_2) \;\middle|\; y_1 + 4y_2 \le \frac{25}{2} \right\}, \; \Omega_2^1 = [0,5]^2 \cap \left\{ (y_1, y_2) \;\middle|\; 5y_1 + 2y_2 \le \frac{35}{2} \right\},$$
$$\Omega_3^1 = [0,5]^2 \cap \{ (y_1, y_2) \mid 3y_1 + 5y_2 \le 20 \}, \; \Omega_4^1 = [0,5]^2 \cap \left\{ (y_1, y_2) \;\middle|\; 7y_1 + 4y_2 \le \frac{55}{2} \right\}.$$

Step 5, 6. Update $k = 1$ and go to Step 2-1.

Similar to the one-dimensional example, the first evaluation point is also shared for all indices. Therefore, the first inner loop is straightforward. The main difference between one-dimensional problems and multi-dimensional ones emerges in Step 4. In one-dimensional problems, the updated localization sets are either left or right side of the segment. However, in multi-dimensional problems, all localization sets are generally different. Therefore, which companions survive is unclear until the next evaluation point is detected.

Step 2-1. The subproblem of KCPM becomes

$$\begin{aligned} \text{minimize} \quad & r \\ \text{subject to} \quad & y_1 + 4y_2 \le r, \\ & 0 \le y_j \le 5, \; j = 1,2. \end{aligned}$$

It returns $\boldsymbol{y}_1^2 = (0,0) \in \Omega_1^1$.

Step 2-2. Since $S_1^1 = \{1,2,3,4\}$, $k_1 = k_2 = k_3 = k_4 = 1$, and $\boldsymbol{y}_1^2 \in \Omega_s^1$ for all $s = 1,2,3,4$, we set $S_1^2 = \{1,2,3,4\}$.

Step 2-3 to 2-5. Set $\boldsymbol{y}_2^2 = \boldsymbol{y}_3^2 = \boldsymbol{y}_4^2 = \boldsymbol{y}_1^2 = (0,0)$ and $S_2^2 = S_3^2 = S_4^2 = S_1^2 = \{1,2,3,4\}$. Update $k_1 = k_2 = k_3 = k_4 = 1 + 1 = 2$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_1^2, g(\boldsymbol{x}) \rangle = -x_1 - 2x_2 - 3x_3 - 4x_4$, we have $\boldsymbol{x}_1^2 = (1,1,1,1)$.

Step 4. Since $g(\boldsymbol{x}_1^2) = (8,6)$, we have

$$\Omega_1^2 = \Omega_1^1 \cap \{ (y_1, y_2) \mid -7y_1 - 2y_2 \leq 0 \}, \ \Omega_2^2 = \Omega_2^1 \cap \{ (y_1, y_2) \mid -3y_1 - 4y_2 \leq 0 \},$$
$$\Omega_3^2 = \Omega_3^1 \cap \{ (y_1, y_2) \mid -5y_1 - y_2 \leq 0 \}, \ \Omega_4^2 = \Omega_4^1 \cap \{ (y_1, y_2) \mid -y_1 - 2y_2 \leq 0 \}.$$

Step 5, 6. Update $k = 2$ and go to Step 2-1.

Contrary to one-dimensional problems, the result of Step 2-2 is generally not predictable at previous Step 4. In this inner loop, we can confirm that all companions survive.

Step 2-1. A new constraint $-7y_1 - 2y_2 + 10 \leq r$ is added to the subproblem of KCPM, and it returns $\boldsymbol{y}_1^3 = (\frac{5}{4}, 0) \in \Omega_1^2$.

Step 2-2. Since $S_1^2 = \{1,2,3,4\}$, $k_1 = k_2 = k_3 = k_4 = 2$, and $\boldsymbol{y}_1^3 \in \Omega_s^2$ for all $s = 1,2,3,4$, we set $S_1^3 = \{1,2,3,4\}$.

Step 2-3 to 2-5. Set $\boldsymbol{y}_2^3 = \boldsymbol{y}_3^3 = \boldsymbol{y}_4^3 = \boldsymbol{y}_1^3 = (\frac{5}{4}, 0)$ and $S_2^3 = S_3^3 = S_4^3 = S_1^3 = \{1,2,3,4\}$. Update $k_1 = k_2 = k_3 = k_4 = 2 + 1 = 3$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_1^3, g(\boldsymbol{x}) \rangle = \frac{3}{2}x_1 + \frac{1}{2}x_2 - \frac{1}{2}x_3 - \frac{3}{2}x_4$, we have $\boldsymbol{x}_1^3 = (0,0,1,1)$.

Step 4. Since $g(\boldsymbol{x}_1^3) = (4,6)$, we have

$$\Omega_1^3 = \Omega_1^2 \cap \left\{ (y_1, y_2) \;\middle|\; -3y_1 - 2y_2 \leq -\frac{15}{4} \right\}, \ \Omega_2^3 = \Omega_2^2 \cap \left\{ (y_1, y_2) \;\middle|\; y_1 - 4y_2 \leq \frac{5}{4} \right\},$$
$$\Omega_3^3 = \Omega_3^2 \cap \left\{ (y_1, y_2) \;\middle|\; -y_1 - y_2 \leq -\frac{5}{4} \right\}, \ \Omega_4^3 = \Omega_4^2 \cap \left\{ (y_1, y_2) \;\middle|\; 3y_1 - 2y_2 \leq \frac{15}{4} \right\}.$$

Step 5, 6. Update $k = 3$ and go to Step 2-1.

Step 2-1. A new constraint $-3y_1 - 2y_2 + 7 \leq r$ is added to the subproblem, and it returns $\boldsymbol{y}_1^4 = (\frac{7}{4}, 0) \in \Omega_1^3$.

Step 2-2. We have $S_1^3 = \{1,2,3,4\}$, $k_1 = k_2 = k_3 = k_4 = 3$, and $\boldsymbol{y}_1^4 \in \Omega_s^3$ for $s = 1,3$. However, $\boldsymbol{y}_1^4 \notin \Omega_s^3$ for $s = 2,4$. Therefore, we set $S_1^4 = \{1,3\}$.

Step 2-3 to 2-5. Set $\boldsymbol{y}_3^4 = \boldsymbol{y}_1^4 = (\frac{7}{4}, 0)$ and $S_3^4 = S_1^4 = \{1,3\}$. Update $k_1 = k_3 = 3 + 1 = 4$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_1^4, g(\boldsymbol{x}) \rangle = \frac{5}{2}x_1 + \frac{3}{2}x_2 + \frac{1}{2}x_3 - \frac{1}{2}x_4$, we have $\boldsymbol{x}_1^4 = (0,0,0,1)$.

Step 4. Since $g(\boldsymbol{x}_1^4) = (2,3)$, we have

$$\Omega_1^4 = \Omega_1^3 \cap \left\{ (y_1, y_2) \;\middle|\; -y_1 + y_2 \leq -\frac{7}{4} \right\}, \ \Omega_3^4 = \Omega_3^3 \cap \left\{ (y_1, y_2) \;\middle|\; y_1 + 2y_2 \leq \frac{7}{4} \right\}.$$

Step 5, 6. Update $k = 4$ and go to Step 2-1.

Step 2-1. A new constraint $-y_1 + y_2 + 4 \leq r$ is added to the subproblem, and it returns $\boldsymbol{y}_1^5 = (2,0) \in \Omega_1^4$.

Step 2-2. We have $S_1^4 = \{1,3\}$, $k_1 = k_3 = 4$. However, $\boldsymbol{y}_1^5 \notin \Omega_3^4$. Therefore, we set $S_1^5 = \{1\}$.

Step 2-3 to 2-5. Update $k_1 = 4 + 1 = 5$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_1^4, g(\boldsymbol{x}) \rangle = 3x_1 + 2x_2 + x_3$, both $(0,0,0,0)$ and $(0,0,0,1)$ are optimal. We set $\boldsymbol{x}_1^5 = (0,0,0,0)$.

Step 4. Since $g(\boldsymbol{x}_1^5) = (0,0)$, we have

$$\Omega_1^5 = \Omega_1^4 \cap \{ (y_1, y_2) \mid y_1 + 4y_2 \leq 2 \}.$$

Step 5, 6. Update $k = 5$ and go to Step 2-1.

Step 2-1. A new constraint $y_1 + 4y_2 \le r$ is added to the subproblem. However, the constraint is equivalent to the one obtained at the second inner iteration. Therefore, the subproblem returns the same solution $\boldsymbol{y}_1^6 = (2, 0) \in \Omega_1^5$.

Step 2-2. We have $S_1^5 = \{1\}$ and $k_1 = 5$. Therefore, we set $S_1^6 = \{1\}$.

Step 2-3 to 2-5. Update $k_1 = 5 + 1 = 6$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_1^5, g(\boldsymbol{x}) \rangle = 3x_1 + 2x_2 + x_3$, both $(0, 0, 0, 0)$ and $(0, 0, 0, 1)$ are optimal. We set $\boldsymbol{x}_1^6 = (0, 0, 0, 0)$.

Step 4. Since $g(\boldsymbol{x}_1^6) = (0, 0)$, we have

$$\Omega_1^6 = \Omega_1^5 \cap \{\, (y_1, y_2) \mid y_1 + 4y_2 \le 2 \,\} = \Omega_1^5.$$

Step 5. Since $\boldsymbol{y}_1^6 = \boldsymbol{y}_1^5$, we regard that Ω_1^6 is sufficiently small. Therefore, we go to Step 7.

Step 7. Set $t = 1 + 1 = 2$, and go to Step 1.

In Step 2-1, a new constraint $y_1 + 4y_2 \le r$ that depends on $\boldsymbol{x}_1^5 = (0, 0, 0, 0)$. Let us consider what happens if $\boldsymbol{x}_1^5 = (0, 0, 0, 1)$ is selected. In this case, we have $g(\boldsymbol{x}_1^5) = (2, 3)$. It follows that the newly added constraint becomes $-y_1 + y_2 + 4 \le r$. It is equivalent to the one obtained at the fourth inner iteration. Therefore, the subproblem of KCPM returns the same solution in either case.

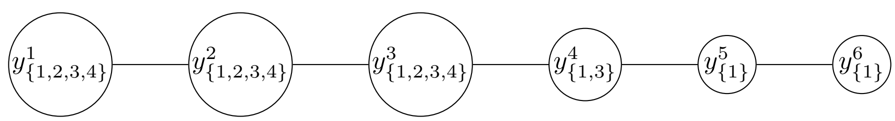

Figure 9: The tree of evaluation points after the first outer loop.

Step 1. Set $k = k_2 = 3$.

Step 2-1. The subproblem of KCPM becomes

$$\begin{aligned} \text{minimize} \quad & r \\ \text{subject to} \quad & 5y_1 + 2y_2 \le r, \\ & -3y_1 - 4y_2 + 10 \le r, \\ & y_1 - 4y_2 + 7 \le r, \\ & 0 \le y_j \le 5,\ j = 1, 2. \end{aligned}$$

It returns $\boldsymbol{y}_2^4 = (0, \frac{5}{3}) \in \Omega_2^3$.

Step 2-2. Since $S_2^3 = \{1, 2, 3, 4\}$, $k_2 = k_4 = 3$, and $\boldsymbol{y}_2^4 \in \Omega_s^3$ for $s = 2, 4$, we set $S_2^4 = \{2, 4\}$.

Step 2-3 to 2-5. Set $\boldsymbol{y}_4^4 = \boldsymbol{y}_2^4 = (0, \frac{5}{3})$ and $S_2^4 = S_4^4 = \{2, 4\}$. Update $k_2 = k_4 = 3 + 1 = 4$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_2^4, g(\boldsymbol{x}) \rangle = -x_1 - 2x_2 + 2x_3 + x_4$, we have $\boldsymbol{x}_2^4 = (1, 1, 0, 0)$.

Step 4. Since $g(\boldsymbol{x}_2^4) = (4, 0)$, we have

$$\Omega_2^4 = \Omega_2^3 \cap \left\{ (y_1, y_2) \;\middle|\; y_1 + 2y_2 \le \frac{10}{3} \right\},\ \Omega_4^4 = \Omega_4^3 \cap \left\{ (y_1, y_2) \;\middle|\; 3y_1 + 4y_2 \le \frac{20}{3} \right\}.$$

Step 5, 6. Update $k = 4$ and go to Step 2-1.

We can construct the subproblem of KCPM in Step 2-1 from the sequence of evaluation points $\{\boldsymbol{y}_2^\ell\}_{\ell=1}^3$ and their evaluations that have been obtained so far.

Step 2-1. A new constraint $y_1+2y_2+3 \le r$ is added to the subproblem, and it returns $\boldsymbol{y}_2^5 = (\frac{3}{4}, \frac{2}{3}) \in \Omega_2^4$.

Step 2-2. Since $S_2^4 = \{2,4\}$, $k_2 = k_4 = 4$, and $\boldsymbol{y}_2^5 \in \Omega_s^4$ for $s = 2, 4$, we set $S_2^5 = \{2,4\}$.

Step 2-3 to 2-5. Set $\boldsymbol{y}_4^5 = \boldsymbol{y}_2^5 = (\frac{3}{4}, \frac{2}{3})$ and $S_2^5 = S_4^5 = \{2,4\}$. Update $k_2 = k_4 = 4+1 = 5$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_2^5, g(\boldsymbol{x})\rangle = \frac{1}{2}x_1 - \frac{1}{2}x_2 + \frac{1}{2}x_3 - \frac{1}{2}x_4$, we have $\boldsymbol{x}_2^5 = (0,1,0,1)$.

Step 4. Since $g(\boldsymbol{x}_2^5) = (4,3)$, we have

$$\Omega_2^5 = \Omega_2^4 \cap \left\{ (y_1, y_2) \;\middle|\; y_1 - y_2 \le \frac{1}{12} \right\},\ \Omega_4^5 = \Omega_4^4 \cap \left\{ (y_1, y_2) \;\middle|\; 3y_1 + y_2 \le \frac{35}{12} \right\}.$$

Step 5, 6. Update $k = 5$ and go to Step 2-1.

Step 2-1. A new constraint $y_1 - y_2 + 6 \le r$ is added to the subproblem, and it returns $\boldsymbol{y}_2^6 = (\frac{1}{4}, 1) \in \Omega_2^5$.

Step 2-2. Since $S_2^5 = \{2,4\}$, $k_2 = k_4 = 5$, and $\boldsymbol{y}_2^6 \in \Omega_s^5$ for $s = 2, 4$, we set $S_2^6 = \{2,4\}$.

Step 2-3 to 2-5. Set $\boldsymbol{y}_4^6 = \boldsymbol{y}_2^6 = (\frac{1}{4}, 1)$ and $S_2^6 = S_4^6 = \{2,4\}$. Update $k_2 = k_4 = 6$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_2^6, g(\boldsymbol{x})\rangle = -\frac{1}{2}x_1 - \frac{3}{2}x_2 + \frac{1}{2}x_3 - \frac{1}{2}x_4$, we have $\boldsymbol{x}_2^6 = (1,1,0,1)$.

Step 4. Since $g(\boldsymbol{x}_2^6) = (6,3)$, we have

$$\Omega_2^6 = \Omega_2^5 \cap \left\{ (y_1, y_2) \;\middle|\; -y_1 - y_2 \le -\frac{5}{4} \right\},\ \Omega_4^6 = \Omega_4^5 \cap \left\{ (y_1, y_2) \;\middle|\; y_1 + y_2 \le \frac{5}{4} \right\}.$$

Step 5, 6. Update $k = 6$ and go to Step 2-1.

Step 2-1. A new constraint $-y_1 - y_2 + 7 \le r$ is added to the subproblem, and it returns $\boldsymbol{y}_2^7 = (\frac{1}{2}, 1) \in \Omega_2^6$.

Step 2-2. We have $S_2^6 = \{2,4\}$ and $k_2 = k_4 = 6$. However, $\boldsymbol{y}_2^6 \notin \Omega_4^6$. Therefore, we set $S_2^7 = \{2\}$.

Step 2-3 to 2-5. Update $k_2 = 7$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_2^7, g(\boldsymbol{x})\rangle = -x_2 + x_3$, four minimizers $(0,1,0,0), (0,1,0,1), (1,1,0,0), (1,1,0,1)$ are possible. Among them, we set $\boldsymbol{x}_2^7 = (0,1,0,0)$.

Step 4. Since $g(\boldsymbol{x}_2^7) = (2,0)$, we have

$$\Omega_2^7 = \Omega_2^6 \cap \left\{ (y_1, y_2) \;\middle|\; 3y_1 + 4y_2 \le -\frac{5}{4} \right\}.$$

Step 5, 6. Update $k = 7$ and go to Step 2-1.

In Step 4, we choose $\boldsymbol{x}_2^7 = (0,1,0,0)$ among the four candidates. Let us consider what happens if we choose another candidate. Since the other three candidates $(0,1,0,1), (1,1,0,0), (1,1,0,1)$ have already been obtained at previous iterations, they do not generate new constraints. Therefore, Step 2-1 in the next iteration returns the same evaluation point.

Step 2-1. A new constraint $3y_1 + 2y_2 + 2 \le r$ is added to the subproblem, and it returns $\boldsymbol{y}_2^8 = (\frac{1}{2}, 1) \in \Omega_2^7$ that is equivalent to $\boldsymbol{y}_2^7$.

Step 2-2. We have $S_2^7 = \{2\}$ and $k_2 = 7$. Therefore, we set $S_2^8 = \{2\}$.

Step 2-3 to 2-5. Update $k_2 = 8$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_2^7, g(\boldsymbol{x})\rangle = -x_2 + x_3$, four minimizers $(0,1,0,0), (0,1,0,1), (1,1,0,0), (1,1,0,1)$ are possible. Among them, we set $\boldsymbol{x}_2^8 = (0,1,0,1)$.

Step 4. Since $g(\boldsymbol{x}_2^8) = (2,3)$, we have

$$\Omega_2^8 = \Omega_2^7 \cap \left\{ (y_1, y_2) \;\middle|\; 3y_1 - y_2 \leq \frac{1}{2} \right\}.$$

Step 5. Since $\boldsymbol{y}_2^8 = \boldsymbol{y}_2^7$, we regard that Ω_2^8 is sufficiently small, and we go to Step 7.

Step 7. Set $t = 2 + 1 = 3$, and go to Step 1.

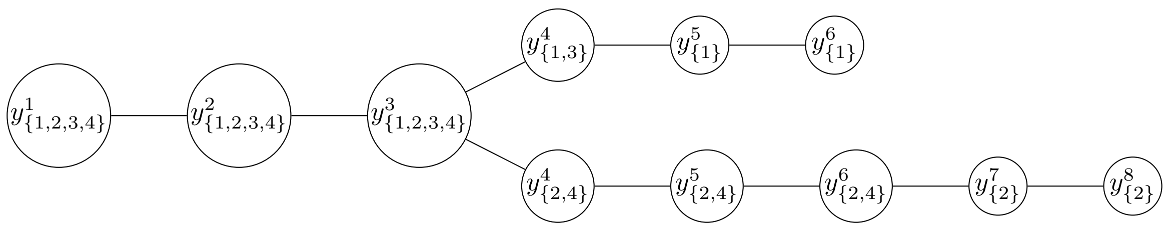

Figure 10: The tree of evaluation points after the second outer loop.

Step 1. Set $k = k_3 = 4$.

Step 2-1. The subproblem of KCPM becomes

$$\begin{aligned}
\text{minimize} \quad & r \\
\text{subject to} \quad & 3y_1 + 5y_2 \leq r, \\
& -5y_1 - y_2 + 10 \leq r, \\
& -y_1 - y_2 + 7 \leq r, \\
& y_1 + 2y_2 + 4 \leq r, \\
& 0 \leq y_j \leq 5, \; j = 1, 2.
\end{aligned}$$

It returns $\boldsymbol{y}_3^5 = (\frac{3}{2}, 0) \in \Omega_3^4$.

Step 2-2. Since $S_3^4 = \{1,3\}$ and $k_3 = 4$, we set $S_3^5 = \{3\}$.

Step 2-3 to 2-5. Update $k_3 = 4 + 1 = 5$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_3^5, g(\boldsymbol{x})\rangle = 2x_1 + x_2 - x_4$, both $(0,0,0,1), (0,0,1,1)$ are possible. We choose $\boldsymbol{x}_3^5 = (0,0,0,1)$.

Step 4. Since $g(\boldsymbol{x}_3^5) = (2,3)$, we have

$$\Omega_3^5 = \Omega_3^4 \cap \left\{ (y_1, y_2) \;\middle|\; y_1 + 2y_2 \leq \frac{3}{2} \right\}.$$

Step 5, 6. Update $k = 5$ and go to Step 2-1.

We can construct the subproblem of KCPM in Step 2-1 from the sequence of evaluation points $\{\boldsymbol{y}_3^\ell\}_{\ell=1}^4$ and their evaluations that have been obtained so far.

Step 2-1. A new constraint $y_1 + 2y_2 + 4 \leq r$ is added to the subproblem. However, the constraint is equivalent to the one obtained so far. Therefore, the subproblem returns the same solution $\boldsymbol{y}_3^6 = (\frac{3}{2}, 0) \in \Omega_3^5$.

Step 2-2. Since $S_3^5 = \{3\}$ and $k_3 = 5$, we set $S_3^6 = \{3\}$.

Step 2-3 to 2-5. Update $k_3 = 6$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_3^6, g(\boldsymbol{x})\rangle = 2x_1 + x_2 - x_4$ in the same way, both $(0,0,0,1), (0,0,1,1)$ are possible. We choose $\boldsymbol{x}_3^6 = (0,0,1,1)$.

Step 4. Since $g(\boldsymbol{x}_3^6) = (4,6)$, we have

$$\Omega_3^6 = \Omega_3^5 \cap \left\{ (y_1, y_2) \;\middle|\; -y_1 - 2y_2 \le -\frac{3}{2} \right\}.$$

Step 5. Since $\boldsymbol{y}_3^6 = \boldsymbol{y}_3^5$, we regard that Ω_3^6 is sufficiently small, and we go to Step 7.

Step 7. Set $t = 3 + 1 = 4$, and go to Step 1.

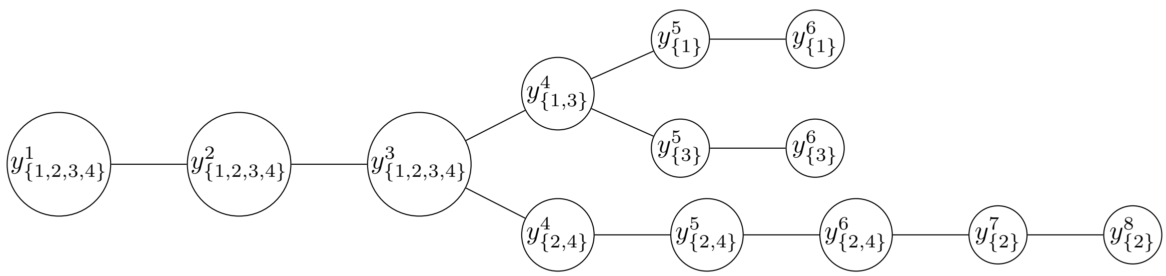

Figure 11: The tree of evaluation points after the third outer loop.

Step 1. Set $k = k_4 = 6$.

Step 2-1. The subproblem of KCPM becomes

$$\begin{array}{rl} \text{minimize} & r \\ \text{subject to} & 7y_1 + 4y_2 \le r, \\ & -y_1 - 2y_2 + 10 \le r, \\ & 3y_1 - 2y_2 + 7 \le r, \\ & 3y_1 + 4y_2 + 3 \le r, \\ & 3y_1 + y_2 + 6 \le r, \\ & y_1 + y_2 + 7 \le r, \\ & 0 \le y_j \le 5, \; j = 1, 2. \end{array}$$

It returns $\boldsymbol{y}_4^7 = (0,1) \in \Omega_4^6$.

Step 2-2. Since $S_4^6 = \{2,4\}$ and $k_4 = 6$, we set $S_4^7 = \{4\}$.

Step 2-3 to 2-5. Update $k_4 = 7$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_4^7, g(\boldsymbol{x})\rangle = -x_1 - 2x_2 - x_4$, both $(1,1,0,1), (1,1,1,1)$ are possible. We choose $\boldsymbol{x}_4^7 = (1,1,0,1)$.

Step 4. Since $g(\boldsymbol{x}_4^7) = (6,3)$, we have

$$\Omega_4^7 = \Omega_4^6 \cap \{ (y_1, y_2) \mid -y_1 - 2y_2 \le -2 \}.$$

Step 5, 6. Update $k = 7$ and go to Step 2-1.

We can construct the subproblem of KCPM in Step 2-1 from the sequence of evaluation points $\{\boldsymbol{y}_4^\ell\}_{\ell=1}^6$ and their evaluations that have been obtained so far.

Step 2-1. A new constraint $y_1 + y_2 + 7 \leq r$ is added to the subproblem. However, the constraint is equivalent to the one obtained so far. Therefore, the subproblem returns the same solution $\boldsymbol{y}_4^8 = (0, 1) \in \Omega_4^7$.

Step 2-2. Since $S_4^7 = \{4\}$ and $k_4 = 7$, we set $S_4^7 = \{4\}$.

Step 2-3 to 2-5. Update $k_4 = 8$.

Step 3. Since $f(\boldsymbol{x}) + \langle \boldsymbol{y}_4^8, g(\boldsymbol{x})\rangle = -x_1 - 2x_2 - x_4$ in the same way, both $(1, 1, 0, 1), (1, 1, 1, 1)$ are possible. We choose $\boldsymbol{x}_4^8 = (1, 1, 1, 1)$.

Step 4. Since $g(\boldsymbol{x}_4^8) = (8, 6)$, we have

$$\Omega_4^8 = \Omega_4^7 \cap \left\{ (y_1, y_2) \;\middle|\; y_1 + 2y_2 \leq \frac{3}{2} \right\}.$$

Step 5. Since $\boldsymbol{y}_4^8 = \boldsymbol{y}_4^7$, we regard that Ω_3^6 is sufficiently small, and we go to Step 7.

Step 7. Since $t = 4 = T$, we terminate the algorithm.

A.2 Proof of Lemma 4.1

Proof. We first consider the dual problem of the subproblem of KCPM. Let $\lambda \in \mathbb{R}_+^k$ be a dual vector that corresponds to cutting planes constraints (2.7), $(\mu^-, \mu^+) \in \mathbb{R}_+^m \times \mathbb{R}_+^m$ be a pair of dual vectors corresponding to lower and upper bound constraints. Then the dual problem can be written as follows:

$$\begin{aligned}
\text{maximize} \quad & \sum_{\ell=1}^k \lambda_\ell \left\{ q_u(y^\ell) - \langle u - g(x^\ell), y^\ell \rangle \right\} - \langle \bar{y}, \mu^+ \rangle \\
\text{subject to} \quad & \sum_{\ell}^k \lambda_\ell = 1, \\
& \sum_{\ell=1}^k \lambda_\ell (u - g(x^\ell)) + \mu^+ \geq 0, \qquad\qquad \text{(A.1)} \\
& \lambda \geq 0,\ \mu^+ \geq 0.
\end{aligned}$$

Note that inequality (A.1) is equivalent to $\sum_{\ell=1}^k \lambda_\ell (u - g(x^\ell)) + \mu^+ - \mu^- = 0$ and $\mu^- \geq 0$. We show that the analytic center of Ω^k provides a feasible solution of the dual problem. Let y^{k+1} be the arbitrary analytic center of Ω^k. Then we have the following by the first order optimality condition of (2.8).

$$-\sum_{\ell=1}^k \frac{(u - g(x^\ell))_j}{\langle u - g(x^\ell), y^{k+1} - y^\ell \rangle} - \frac{1}{y_j^{k+1}} + \frac{1}{\bar{y}_j - y_j^{k+1}} = 0,\ j = 1, \ldots, m.$$

Therefore, following pair (λ, μ^+) is a feasible solution of the dual problem.

$$\lambda_\ell = \frac{\tau_\ell}{\langle \mathbf{1}, \tau \rangle}, \quad \mu_j^+ = \frac{1}{\bar{y}_j - y_j^{k+1}} \frac{1}{\langle \mathbf{1}, \tau \rangle},$$

where τ is defined as

$$\tau_\ell = -\frac{1}{\langle u - g(x^\ell), y^{k+1} - y^\ell \rangle}.$$

We also have

$$q_u(y^\ell) - \langle u - g(x^\ell), y^\ell \rangle = -\min_{x \in X}\{f(x) + \langle y^\ell, g(x) \rangle\} + \langle g(x^\ell), y^\ell \rangle = f(x^\ell).$$

Then, it is straightforward that the value of the objective for the feasible solution becomes $\hat{L}_u$. Therefore, weak duality theorem provides us the desired result:

$$\hat{L}_u \leq L_u.$$

□